EXHIBIT E
PART 2

Ignore all outgoing traffic while monitoring the Web server. Modify the destination address to the IP address to that of the Web server. The source port in the configuration is set to 0, indicating connections coming from any source port will be logged. The Message value is set to Web. Priority is set to 2 (Medium). Connection establishment on the GUI is viewable real-time, since the Display action is enabled.

In the features.cfg file, the HTTPGetDecodeCheck is enabled, and all other checks disabled. The action for this check is set to Display and to log to a file.

Sample Configuration 2: Exploit Finder

exp-features.cfg
exp-filter.cfg

This configuration displays all attempts of known exploits against any machine on the network, or originating from any machine on the network. This immediately allows an administrator to check for a break-in, using one of the many ways **RealSecure** checks. **RealSecure** also detects if anyone inside the network is attempting to break-into other machines on the Internet.

To configure this, set up the features configuration to enable all of the exploit checks, while disabling all the simple informational decodes. The filter configuration is then set to filter out services the exploits are attempted against, in this case HTTP, SMTP, FTP, Rlogin, and Ident. Consider configuring the individual features to page or flag some other alert when detecting one of these exploits. These features will indicate a break-in attempt.

Sample Configuration 3: TCP Traffic

all-features.cfg
all-filter.cfg

This configuration displays all TCP traffic going through the network and displays all decodes and vulnerability attempts. This configuration is good for a small network having an active system administrator. Watching **RealSecure**, the administrator can monitor all of the TCP traffic on the network, users and passwords being attempted to each machine files being transferred over FTP and URLs being used over HTTP. These attempts are harder to detect on larger networks.

Changing the one line of the filter configuration to point to a specific machine, can significantly reduce the information that pinpoints all of the TCP traffic coming and going from one particular machine. If there is a suspected hacker on a machine, consider this option to monitor the traffic on the machine. If this is the case, consider enabling log in to a file in order to efficiently monitor the activities of the hacker.

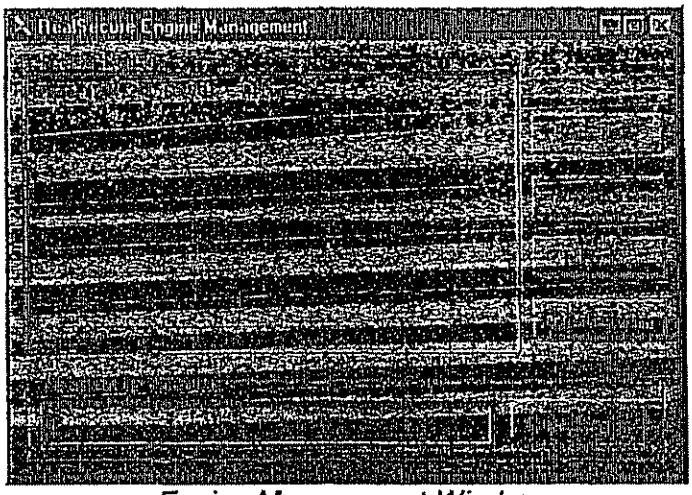
The filter configuration is defined to display all TCP traffic going through the network. Every feature is enabled in the features configuration.

Chapter 4:

Using RealSecure

OVERVIEW

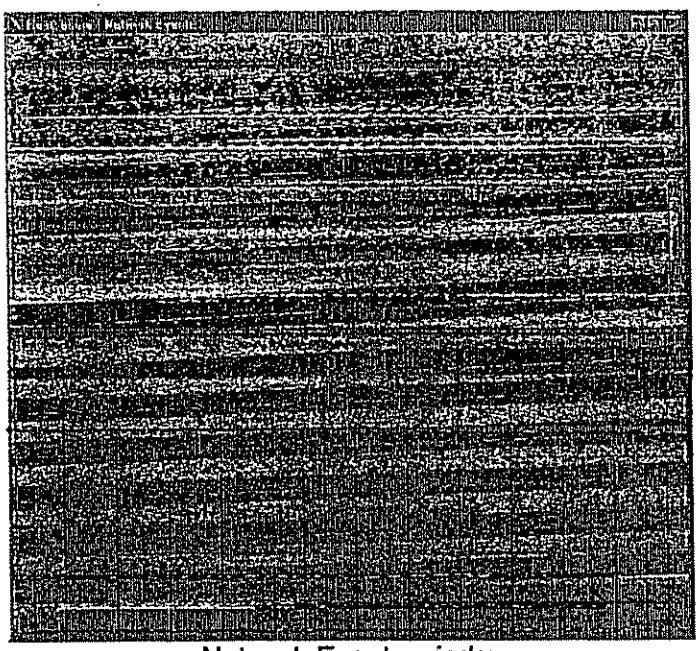
Upon selecting Configure Engines from the RealSecure toolbar, the Engine Management window displays. This window allows for starting, stopping and configuration of engines on each host. If engines are already running on the network, the GUI is automatically contacted and appears within a list on the Engine Management window. The list displays the active engines, one per line as follows...



Engine Management Window

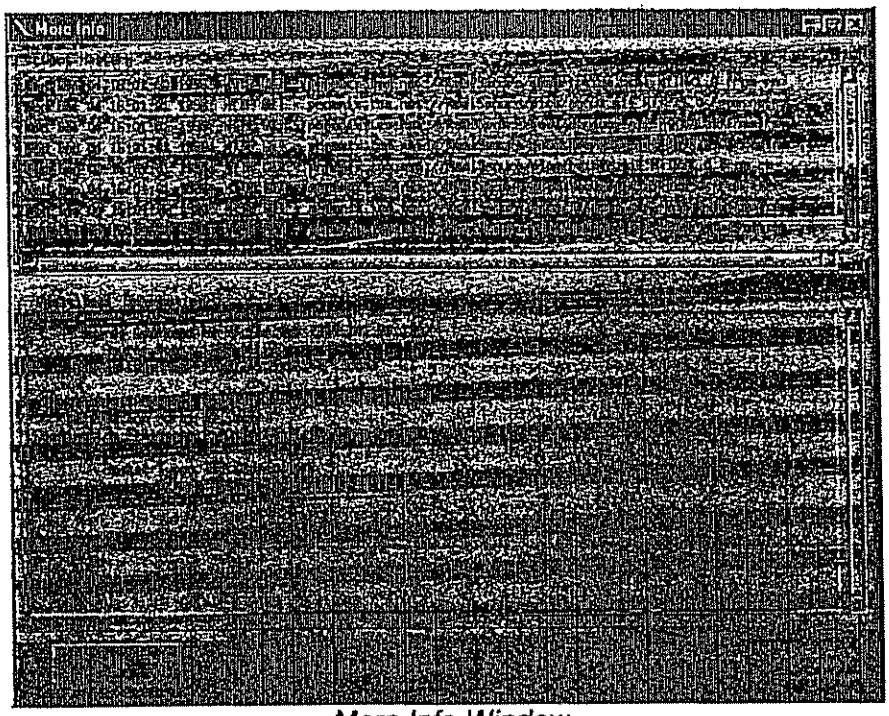
To start an engine, type the name or IP address of a host on which you wish to run an engine and press Enter, or click on Start Engine. After a short delay, the engine will appear in the engine list and the Network Events window will pop up to start displaying events.

Remote engines may only be started on hosts that are configured for running sssd.



Network Events window

As events appear in the window, more information is provided about them by double-clicking on the entry.



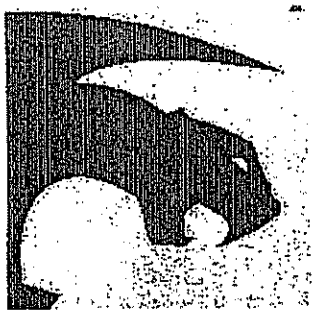
More Info Window

The More Info window displays information about the event, like the Ethernet addresses and TCP/UDP ports associated with the connection. Also, a list of all related events allows you to track a connection from start to finish. To close this window, click on OK. To perform some action with this connection, click on Actions. Note that certain actions are limited by the network state or the underlying protocol. For instance, you can't kill a connection that doesn't exist any more. Double click on the http URL to display the browser with that URL.

The bar at the bottom of the screen displays the number of packets per second seen by the engine(s). If they bar ever goes above 100%, that becomes the new maximum.

Events are added to the top of the list. Old events are moved toward the bottom. Each window (high, medium, and low) has a timeout value for its events (configurable in General Configuration). When the time expires for an event, the event is removed from the screen. If it was logged to a file, you still have a record of the event that will show up on any reports you generate. If the event occurs again after being removed, the new event will be displayed.

As events occur, there may be some that are not desired for viewing. In this case, return to the configuration screens and set those events to disabled (ignore). Alternatively, in the event some events were ignored which should not have been, modify the configuration accordingly. Be sure to retain a backup copy of the known-good configurations.



RealSecure™

Chapter 5:

Generating Reports

OVERVIEW

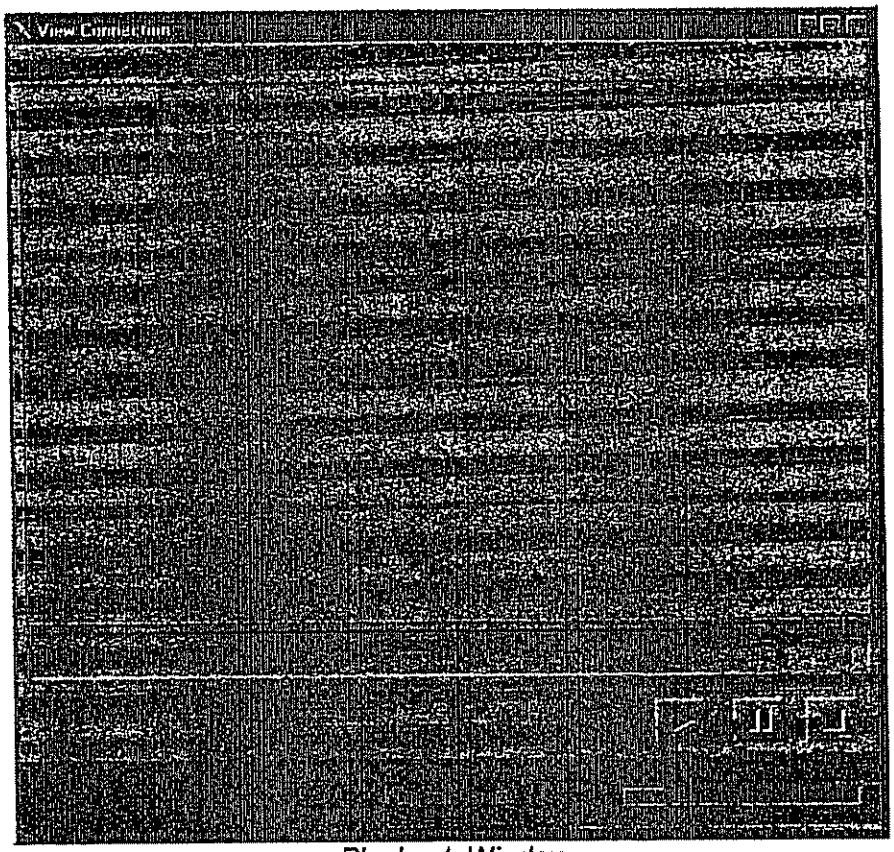
To view a summary of network attacks, utilize the Playback and Reporting features. These reports provide useful information about major network problems and allow for security configuration and modification. Some additional uses include break-in trends and determining corrective action for handling future intrusions.

Playback Feature

Although the Playback Feature a reporting feature *per se*, it does indicate what the intruder saw and typed. To use this feature, obtain a log generated with the Log Raw action and press Enter. Examine the following example:

```
# playback <my-logged-raw-data-file>
```

A list of the connections recorded in that file displays.

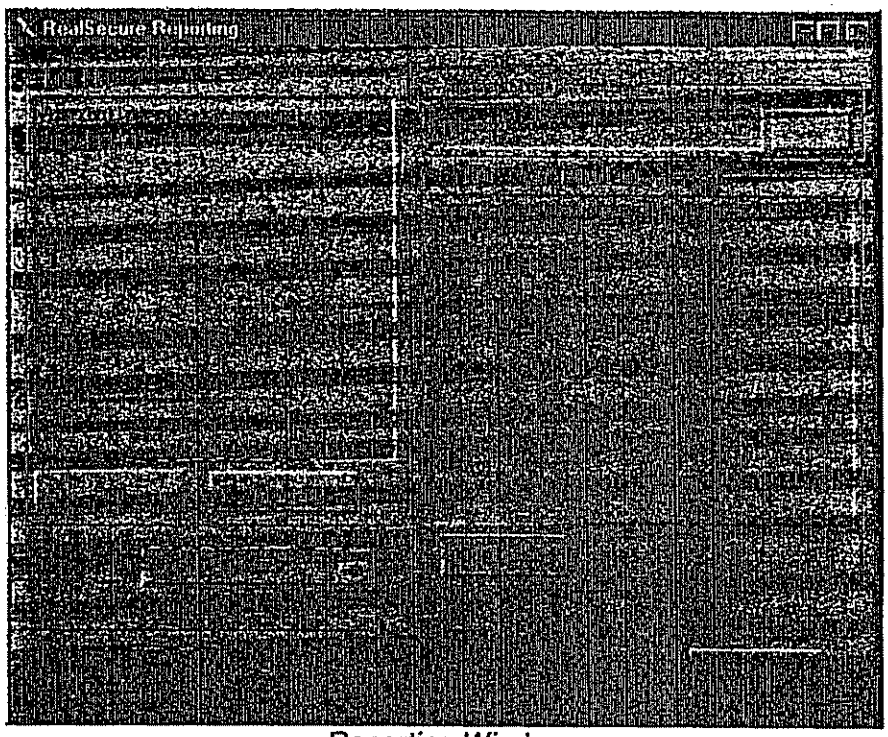


Playback Window

Select the desired report to view and click the **Play** button. The play speed is not modifiable with the slider located at the bottom of the screen. Pressing **Stop** returns to list of connections.

Reporting Feature

The reporting feature uses logs generated with the **Log Info** action. To generate reports, click on the **Reports** button on the **RealSecure** toolbar. The Reporting window displays.



Reporting Window

The Reporting window allows configuration of the log files used to generate a report, where the report will be stored, and settings for generating the report.

On the left of the screen is the list of log files to use to generate a report. To add another file to the list, click on the **Add** button and select the desired file. To remove a file from the list, click on the name in the list and click on **Remove**. The **Report by** menu selects one of the three reports to generate: by source address, destination address and by event type (default).

Click on the ... button to change the destination directory for the reports. RealSecure will create a directory if a selected directory does not exist.

The other list shows what parts of the logs are used when reporting. For instance, you can generate a report on only certain priorities of events, time periods and addresses. To modify this list, click on the **Change** button.

The **Change** button displays a series of settings for report generation. The **Title** option allows for specification of a "Report Generated for..." title in a report. The **Source** and **Destination Addresses** fields allow for specification of an IP or a range of IPs to limit the report. The **Start** and **End Times** fields allow for specification of a reporting time period, in a yymmddhhmm.ss format (year, month, day, hour, minute and second). Examine the following valid configuration:

```
Report generated for:  Internet Security Systems
Start date: 9610250900
End date: 9611250900
Source addresses: 10.0.0.1,20.0.0.1-20.0.0.120
Destination addresses: Ken,Barbie
```

The example provided, generates a report on all events from October 25, 1996 at 9:00 a.m. to November 25, 1996 at 9:00 a.m. Only events coming from the host 10.0.0.1 and hosts 20.0.0.1 to 20.0.0.120 that are destined for the two hosts Ken and Barbie will display.

After setting up a report, click on **Generate** to generate a report. After a short delay, a browser will pop up on the report title page. If an error has occurred, a system error message is delivered to the desktop explaining why.

(This page intentionally left blank)

RealSecure™

Appendix A:

Features and Attack Signatures

OVERVIEW

This appendix summarizes RealSecure's features and the attack signatures it checks.

- IP Fragmentation
- Ping Flooding
- ARP Check
- IP Duplicate Check
- IP Half Scan
- IP Unknown Protocol
- UDP Bomb
- SYN Flood
- Source Routing
- Satan Vulnerability Check
- Chargen Denial of Service Vulnerability Check
- Echo Denial of Service Vulnerability Check
- TFTP Get Vulnerability Check
- TFTP Put Vulnerability Check

- Rwhod Vulnerability Check
- Finger User Decode
- Finger Bomb Vulnerability Check
- RTM Finger Vulnerability Check
- Rlogin Decoding
- Rlogin -froot Vulnerability Check
- FTP Username Decoding
- FTP Password Decoding
- FTP Site Command Decoding
- FTP GET File Decoding
- FTP PUT File Decoding
- FTP Mkdir Decoding
- FTP CWD ~root Vulnerability Check
- HTTP GET Decoding
- HTTP PHF Vulnerability Check
- HTTP Test-Cgi Vulnerability Check
- HTTP..Vulnerability Check
- HTTP Authentication Decode
- HTTP Java Decoding
- HTTP IIS 3.0 Asp Dot Vulnerability Check
- HTTP IIS 3.0 Asp 2E Vulnerability Check
- HTTP Internet Explorer 3.0 .URL/.LNK Vulnerability Check
- Ident User Decoding
- Ident Buffer Overflow Vulnerability Check
- Ident Newline Vulnerability Check
- POP Username Decoding
- POP Password Decoding

- **RSH Decoding**
- **E-Mail From**
- **E-Mail To**
- **E-Mail Subject**
- **E-Mail VRFY**
- **E-Mail EXPN**
- **E-Mail WIZ Vulnerability Check**
- **E-Mail DEBUG Vulnerability Check**
- **E-Mail Pipe Vulnerability Check**
- **E-Mail Decode Vulnerability Check**
- **IRC Nick Decode**
- **IRC Channel Decode**
- **IRC Message Decode**
- **NNTP Username Decoding**
- **NNTP Password Decoding**
- **NNTP Group Decoding**
- **Talk Request Decoding**
- **Talk Flash Vulnerability Check**
- **HP/UX RemoteWatch Vulnerability Check**

IP Fragmentation

An IP packet is sometimes split into several fragments when it is transmitted over the network. These fragments are then reassembled at the destination to form a full IP packet. Some routers that filter out packets based on information in the TCP header rely on the information in the first fragment, then blindly pass the remaining fragments. It is possible to construct individual fragments of an IP packet so that subsequent packets overlap. As a result, overwrite parts of the TCP header when they are reassembled at the destination. The result of this is that an

intermediate filtering router is tricked into believing that a packet is destined for an allowed service. In reality, the packet is destined for a service that would normally be filtered.

Ping Flooding

A Ping Flood is an attempt to saturate a network with packets in order to slow or stop legitimate traffic going through the network. A continuous series of ICMP Echo Requests are made to a target host on the network, which then responds with an ICMP Echo Reply. The continuing combination of requests and replies slow the network and cause legitimate traffic to continue at a significantly reduced speed or, in extreme cases, to disconnect.

ARP Check

ARP, Address Resolution Protocol, is used to determine the Ethernet address of a machine on a network given its IP address. If an ARP is received for a machine on the network, it immediately sends a reply. If the machine the ARP is destined for has crashed or otherwise disconnected from the network, several ARPs will be sent to it without any response. This lack of response to ARP packets is used to determine if a machine on the network has crashed.

IP Duplicate Check

Only one machine on a network should send packets with a specific IP address. If a second machine on the network starts to send packets claiming to have the same source address, a network problem has occurred. A machine on the network may be misconfigured to have the same IP address as another machine, causing network conflicts. The other possibility, is that a machine on the network may be sending out IP packets with a forged source address.

IP Half Scan

A standard TCP connection is established by sending a SYN packet to the destination host. If the destination is waiting for a connection on the specified port, it will respond with a SYN/ACK packet. The initial sender then replies to the SYN/ACK with an ACK packet, and the connection is established. If the destination host is not waiting for a connection on the specified port, it will respond with an RST packet instead of a SYN/ACK. Most system logs do not log completed connections until the final ACK packet is received from the source. Sending an RST packet instead of the final ACK results in the connection never actually being established; so no logging takes place. Because the source can identify whether the destination host sent a SYN/ACK or an RST, an attacker can determine exactly what ports are open for connections, without the destination ever being aware of probing.

IP Unknown Protocol

A standard IP packet contains an 8-bit protocol field. Common values for this field include 6 (TCP), 17 (UDP), and 1 (ICMP). Attackers sometimes use a non-standard value for this field, in order to exchange data between machines without logging mechanisms detecting the data that is being transmitted.

UDP Bomb

A UDP packet that is constructed with illegal values in certain fields will cause some older operating systems to crash when the packet is received. If the target machine does crash, it is often difficult to determine the cause. Most operating systems that are not vulnerable to this problem will silently discard the invalid packet, leaving no traces that it was being subjected to a malicious attack.

SYN Flood

A standard TCP connection is established by sending a SYN packet to the destination host. If the destination is waiting for a connection on the specified port, it will respond with a SYN/ACK packet. The initial sender then replies to the SYN/ACK with an ACK packet, and the connection is established. When the SYN/ACK is sent back to the source, a block of memory is allocated to hold information about the state of the connection that is currently being established. Until the final ACK is received or a timeout is reached, this block of memory sits unused, waiting for more information to be received from the source host. By sending numerous SYN packets to a host, the destination will exhaust the portion of memory it has on-hand to deal with opening connections. Legitimate connections will no longer be able to connect to the host. This situation can be detected by the flood of SYN packets without accompanying responses. It can be corrected by sending the destination RST packets that correspond to the initial SYNs. This results in the destination host freeing up that block of memory and making room for a new legitimate connection.

Source Routing

IP packets sent over the Internet are normally sent between different routers, in order to reach their final destination. The route each packet takes is determined dynamically by each router along the way. Enabling the source routing option on an IP packet allows the packet itself to make known to each router, the path it wishes to take to reach its final destination. By routing packets through a path

that bypasses filtering routers and other normal security mechanisms, an attacker may be able to reach a host that normally could not be reached. Also, it can be used to authenticate an intruder to systems that rely on the source IP address for access control.

Satan Vulnerability Check

(requires filter for UDP port 1)
This check will recognize if a Satan normal or heavy scan of your machine is taking place. Satan is a freely available tool that allows someone to scan your machine for services and a small set of common vulnerabilities.

Chargen Denial of Service Vulnerability Check

(requires filter for UDP port 19)
This check will watch for attempts at performing a denial of service attack against a machine on your network by attempting to engage a machine in a chargen flood against itself.

Echo Denial of Service Vulnerability Check

(requires filter for UDP port 7)
This check watches for attempts at performing a denial of service attack against a machine on your network by attempting to engage a machine in an echo flood against itself.

Rwhod Vulnerability Check

(requires filter for UDP port 513)
This check watches for a malformed rwho UDP packet containing a buffer overflow, that can be used by attackers to perform a denial of service attack against the rwho service or to attempt to execute arbitrary code on a remote machine.

Rlogin Decoding

(requires filter for TCP port 513)
An Rlogin connection allows a user to remotely login to a host without a password by using a trust relationship between the account on the source machine and on the destination host. The source machine and username, along with the destination machine and username are logged with this feature.

Rlogin -froot Vulnerability Check

(requires filter for TCP port 513)

If a remote user passes the name -froot to rlogin to a machine, certain operating systems will bypass normal security mechanisms and log in the user directly as root. This vulnerability allows anyone who can access the rlogin service on the target host to gain immediate root access to the machine.

FTP Username Decoding

(requires filter for TCP port 21)

FTP, File Transfer Protocol, allows users to transfer files between machines. Username decoding discovers the name of the account being used to transfer files across the network.

FTP Password Decoding

(requires filter for TCP port 21)

FTP passes a plain text password across the network in order to authenticate that a user has access to the files on the destination host. This password is discovered using FTP password decoding. This allows an administrator to log invalid password attempts, check passwords for strength against attack and keep complete logs of activity.

FTP Site Command Decoding

(requires filter for TCP port 21)

The FTP site command allows a user to execute certain commands on a destination host in addition to the normal FTP facility of transferring files. In ordinary usage of FTP, this is not a commonly used command. While there may be a legitimate reason to execute site commands under certain circumstances, this facility has also been used to gain access. Consequently, an administrator may wish to view and log the site commands being executed to check for possible abuse.

FTP GET File Decoding

(requires filter for TCP port 21)

Files being transferred from the destination host to the source host use a GET command in order to transfer the files. FTP GET

decoding discovers all files that are being transferred to the source host over FTP.

FTP PUT File Decoding

(requires filter for TCP port 21)

Files being transferred from the source host to the destination host use a PUT command in order to transfer the files. FTP PUT decoding discovers all files that are being transferred to the destination host over FTP.

FTP Mkdir Decoding

(requires filter for TCP port 21)

FTP allows a user to create a new directory on the target machine. FTP Mkdir decoding discovers all new directories that are created through FTP.

FTP CWD ~root Vulnerability Check

(requires filter for TCP port 21)

Certain versions of the FTP daemon allow access to files on a machine through a sequence of commands culminating with CWD ~root. This vulnerability allows an attacker who can access FTP on the target host to transfer files that he/she would not normally have access.

HTTP GET Decoding

(requires filter for TCP port 80)

Pages, images, and all other information that is viewed through a Web browser on the World Wide Web are transferred through HTTP using the GET command. HTTP GET decoding discovers all Web pages being transmitted unsecurely to a machine. This allows an administrator to track, log and view Web traffic on the network.

HTTP PHF Vulnerability Check

(requires filter for TCP port 80)

The cgi-bin script PHF, which comes preinstalled with several versions of NCSA and Apache Web servers, contains a vulnerability that allows anyone who can access a Web site to the machine(s).

HTTP Test-Cgi Vulnerability Check

(requires filter for TCP port 80)

This check recognizes an attack on the cgi-bin test-cgi script. This program, installed by default with certain versions of Apache and NCSA web servers, allows a remote attacker to gain information

about the contents of the cgi-bin directory of your web server which can be used for further attacks.

HTTP.. Vulnerability Check

(requires filter for TCP port 80)
This check recognizes an attack to attempt to obtain information above the "ServerRoot" directory. Some web servers vulnerable to this attack will allow remote users to list the contents of any directory on the system using this type of attack.

HTTP Authentication Decode

(requires filter for TCP port 80)
This decode will log the username and password that is being used to authenticate using HTTP Basic authentication to a web server. This authentication uses Base64 encoding and can be used for such purposes as determining what user accounts are logging into web servers from what machines, log brute force attacks against your web server, and to keep general logs of username and password attempts.

HTTP Java Decoding

(requires filter for TCP port 80)
This decoding recognizes when a web browser attempts to obtain a file containing Java bytecode. This should only occur if a user has Java enabled on their web browser.

HTTP IIS 3.0 Asp Dot Vulnerability Check

(requires filter for TCP port 80)
Microsoft's IIS 3.0 server has a security hole that allows execution of code by inserting a '.' After an active server push URL. This check will recognize attempts to exploit this vulnerability.

HTTP IIS 3.0 Asp 2E Vulnerability Check

(requires filter for TCP port 80)
Microsoft's IIS 3.0 server installed with the hot-fix to solve the Asp Dot vulnerability introduced a new security hole that allows viewing

the contents of an active server push URL by using the hexadecimal value '2e' instead of a '.' in the URL name. This check will recognize attempts to exploit this vulnerability to view the contents of pages.

HTTP Internet Explorer 3.0 .URL/.LNK Vulnerability Check

(requires filter for TCP port 80)
Microsoft's Internet Explorer versions 3.0 and 3.01 have a vulnerability which results in a web site being able to execute an arbitrary program on a machine running Microsoft Windows and browsing the web using MSIE. This vulnerability check will detect when a web site attempts to exploit this vulnerability.

Ident User Decoding

(requires filter for TCP port 113)

The Ident port is used by services to identify the account by which a connection is being made on a machine. This can be used to track a connection back to a specific user on a multi-user machine.

Ident Buffer Overflow Vulnerability Check

(requires filter for TCP port 113)

Certain programs that connect back to the ident service to log user information, expect a properly formatted response. If the response is longer than expected, the buffer that the response is read into is overflowed, allowing the remote user to execute commands on the host machine.

Ident Newline Vulnerability Check

(requires filter for TCP port 113)

Certain programs that connect back to the ident service to log user information expect a properly formatted response. If the response contains newlines, the response may be improperly parsed, allowing the remote user to execute commands on the host machine.

POP Username Decoding

(requires filter for TCP port 119)

The POP service is used by numerous e-mail programs to retrieve e-mail from a mail server and read it on a local machine. POP username decoding discovers the username of the user who is reading mail through the POP service.

POP Password Decoding

(requires filter for TCP port 119)

POP password decoding discovers all successful and unsuccessful passwords that a user attempts to use to login to a mail server using POP.

RSH Decoding

(requires filter for TCP port 512)

RSH, the remote shell command, allows a user to execute a shell command over the network using a trust relationship between the user on the local machine and the user account on the remote machine. RSH decoding discovers both the local and remote usernames as well as the command that is being executed.

E-Mail From

(requires filter for TCP port 25)

This decoding discovers the sender of all mail that is sent over the network using SMTP.

E-Mail To

(requires filter for TCP port 25)

This decoding discovers the recipient of all mail that is sent over the network using SMTP.

E-Mail Subject

(requires filter for TCP port 25)

This decoding discovers the subject line of all mail that is sent over the network using SMTP.

E-Mail VRFY

(requires filter for TCP port 25)

The VRFY command is used to verify if a user on a remote system exists. This is sometimes used legitimately to determine if the recipient of a message at the intended destination is able to receive the message. It is also sometimes used to gain information about users on a system by attempting to determine if certain common account names exist on a machine.

E-Mail EXPN

(requires filter for TCP port 25)

The EXPN command is used to expand the address of a user on a remote system. This is sometimes used legitimately to determine the full address of an intended mail recipient. It is also sometimes used to gain information about users on a system by trying to find out if certain common account names exist on a machine.

E-Mail WIZ Vulnerability Check

(requires filter for TCP port 25)

The WIZ command in Sendmail existed to allow access to a machine under certain circumstances. It is no longer present in current versions of Sendmail, but old versions still in use may allow an attacker to gain root access to a machine by using this command.

E-Mail DEBUG Vulnerability Check

(requires filter for TCP port 25)

The DEBUG command in Sendmail existed to allow debugging of a remote Sendmail daemon. It is no longer present in current versions of Sendmail, but old versions still in use allow an attacker to gain root access to a machine by using this command remotely.

E-Mail Pipe Vulnerability Check

(requires filter for TCP port 25)

By inserting a pipe (|) character into certain fields in an e-mail, Sendmail may be forced to execute a command on the remote host. This results in a remote attacker being able to execute commands as root on the machine.

E-Mail Decode Vulnerability Check

(requires filter for TCP port 25)

By sending mail to decode or uudecode alias that is present in some systems, a remote attacker may be able to create or overwrite files on the remote host.

TFTP Get Vulnerability Check

(requires filter for UDP port 69)
This check watches for attempts to transfer files from a machine using the Trivial File Transfer Protocol (TFTP). This protocol is sometimes legitimately used for bootstrapping by diskless workstations, but it is more often used by attackers to attempt to obtain a password file or other critical system files.

TFTP Put Vulnerability Check

(requires filter for UDP port 69)
This check watches for attempts to transfer files to a machine using the Trival File Transfer Protocol (TFTP). This protocol can be used by attackers to transfer critical system files to a host that is being attacked.

Finger User Decode

(requires filter for TCP port 79)
This decode watches for finger attempts and reports the user (or all users if the attempt was aimed at the whole machine) that the finger was aimed at. Finger has a legitimate use, but is also often used by attackers to gain more information about a machine such as account names, real names, and trusted hosts.

Finger Bomb Vulnerability Check

(requires filter for TCP port 79)

This check watches for attempts to perform a denial of service attack against a machine or for redirecting finger attempts across machines. Redirecting finger attempts is often used by an attacker to hide the original source address of a finger attempt.

RTM Finger Vulnerability Check

(requires filter for TCP port 79)
This check watches for a buffer overflow attempt on the finger service that is used by attackers to attempt to gain access to a machine remotely. This vulnerability is named for Robert T. Morris, author of the Internet Worm that originally popularized this vulnerability.

IRC Nick Decode

(requires filter for TCP port 6667)
This decode watches for changes of a user's nickname on Internet Relay Chat.

IRC Channel Decode

(requires filter for TCP port 6667)
This decode watches for channels that are joined by a user on Internet Relay Chat.

IRC Message Decode

(requires filter for TCP port 6667)
This decode watches for messages that are sent out by a user on Internet Relay Chat.

NNTP Username Decoding

(requires filter for TCP port 119)
The NNTP service is used to read, post, and exchange news from a news
server. NNTP user decoding discovers the username of the user who is
reading or posting news through the NNTP service.

NNTP Password Decoding

(requires filter for TCP port 119)
The NNTP service is used to read, post, and exchange news from a news
server. NNTP password decoding discovers the password attempted to
login to the news server in order to read or post news.

NNTP Group Decoding

(requires filter for TCP port 119)
The NNTP service is used to read, post, and exchange news from a news
server. NNTP group decoding discovers the name of the newsgroup that
a user is accessing on the news server.

Talk Request Decoding

(requires filter for UDP port 517 and 518)
The Talk service is used to engage in a real-time chat with a user on a
remote machine. Talk Request decoding discovers the name and machine that
a talk request is being sent to, along with the name and machine of the
person who is originating the talk request.

Talk Flash Vulnerability Check

(requires filter for UDP port 517 and 518)
The talk service allows the user originating a talk request to specify an
arbitrary string to display for the origin of the talk request. If this
string contains a particular escape sequence, it is possible to cause a
temporary denial of service attack by mangling the contents of a user's

screen. This is commonly known as 'flashing' a user.

HP/UX RemoteWatch Vulnerability Check

(requires filter for TCP port 5556)
Certain versions of HP/UX that come with the RemoteWatch package installed
have a vulnerability which allows a remote attacker to execute arbitrary
commands through the RemoteWatch service on the target machine. This
vulnerability check will watch accesses to the RemoteWatch service and
determine if these accesses are attempting to exploit his vulnerability.

(This page intentionally left blank)

-----BEGIN PGP SIGNED MESSAGE-----

CA-92:19

CERT Advisory
December 7, 1992
Keystroke Logging Banner

The CERT Coordination Center has received information from the United States Department of Justice, General Litigation and Legal Advice Section, Criminal Division, regarding keystroke monitoring by computer systems administrators, as a method of protecting computer systems from unauthorized access.

The information that follows is based on the Justice Department's advice to all federal agencies. CERT strongly suggests adding a notice banner such as the one included below to all systems. Sites not covered by U.S law should consult their legal counsel.

The legality of such monitoring is governed by 18 U.S.C. section 2510 et seq. That statute was last amended in 1986, years before the words "virus" and "worm" became part of our everyday vocabulary. Therefore, not surprisingly, the statute does not directly address the propriety of keystroke monitoring by system administrators.

Attorneys for the Department have engaged in a review of the statute and its legislative history. We believe that such keystroke monitoring of intruders may be defensible under the statute. However, the statute does not expressly authorize such monitoring. Moreover, no court has yet had an opportunity to rule on this issue. If the courts were to decide that such monitoring is improper, it would potentially give rise to both criminal and civil liability for system administrators. Therefore, absent clear guidance from the courts, we believe it is advisable for system administrators who will be engaged in such monitoring to give notice to those who would be subject to monitoring that, by using the system, they are expressly consenting to such monitoring. Since it is important that unauthorized intruders be given notice, some form of banner notice at the time of signing on to the system is required. Simply providing written notice in advance to only authorized users will not be sufficient to place outside hackers on notice.

An agency's banner should give clear and unequivocal notice to intruders that by signing onto the system they are expressly consenting to such monitoring. The banner should also indicate to authorized users that they may be monitored during the effort to monitor the intruder (e.g., if a hacker is downloading a user's file, keystroke monitoring will intercept both the hacker's download command and the authorized user's file). We also understand that system administrators may in some cases monitor authorized users in the course of routine system maintenance. If this is the case, the banner should indicate this fact. An example of an appropriate banner might be as follows:

This system is for the use of authorized users only. Individuals using this computer system without authority, or in excess of their authority, are subject to having all of their activities on this system monitored and recorded by system personnel.

In the course of monitoring individuals improperly using this system, or in the course of system maintenance, the activities of authorized users may also be monitored. Anyone using this system expressly consents to such monitoring and is advised that if such monitoring reveals possible evidence of criminal activity, system personnel may provide the evidence of such monitoring to law enforcement officials.

Each site using this suggested banner should tailor it to their precise needs. Any questions should be directed to your organization's legal counsel.

The CERT Coordination Center wishes to thank Robert S. Mueller, III, Scott Charney and Marty Stansell-Gamm from the United States Department of Justice for their help in preparing this Advisory.

If you believe that your system has been compromised, contact the CERT Coordination Center or your representative in FIRST (Forum of Incident Response and Security Teams).

Internet E-mail: cert@cert.org
Telephone: 412-268-7090 (24-hour hotline)

CERT personnel answer 7:30 a.m.-6:00 p.m. EST(GMT-5)/EDT(GMT-4), on call for emergencies during other hours.

CERT Coordination Center
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213-3890

Past advisories, information about FIRST representatives, and other information related to computer security are available for anonymous FTP from cert.org (192.88.209.5).

-----BEGIN PGP SIGNATURE-----
Version: 2.6.2

iQCVAwUBMaMxEnVP+x0t4w7BAQHKiAP/XmfedOnmkaeQyNpaRF+luXnaQNsIcduY
RicrGD7EJhKmrhsTj4P4uIIiSL9Ue+4WHOF38/yte+WDNqpAITI1wus/0h7pUg1k
urZVI+MzEJJgBGuVqmiu/1hxT10ZtUvtLQGsl/kjfud3e/3xLBfJtSeIwYNNRk/H
Xrv8iQr179Y=
=UuaN
-----END PGP SIGNATURE-----